AFRL-OSR-VA-TR-2013-0172

Out-Learning Attackers: A Game Theoretic Approach to Cyber Defense

John Musacchio
Regents of University of California

April 2013
Final Report

AIR FORCE RESEARCH LABORATORY
AF OFFICE OF SCIENTIFIC RESEARCH (AFOSR)
ARLINGTON, VIRGINIA 22203
AIR FORCE MATERIEL COMMAND

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 28-02-2013	**2. REPORT TYPE** Final Performance Report	**3. DATES COVERED** *(From - To)* Feb. 1, 2009 - Nov. 30, 2012

4. TITLE AND SUBTITLE

Out-Learning Attackers: A Game Theoretic Approach to Cyber Defense

5a. CONTRACT NUMBER

5b. GRANT NUMBER

FA9550-09-1-0049

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Musacchio, John
Frazier, Greg
Kreidl, Pat

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Regents of University of California, Santa Cruz
1156 High Street
Santa Cruz, CA 95064

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Office of Scientific Research
875 N. Randolph St. Room 3112
Arlington, VA 22203
Herklotz, Robert

10. SPONSOR/MONITOR'S ACRONYM(S)

AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

AFRL-OSR-VA-TR-2013-0172

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution A: Approved for public release

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In this project we have constructed a Markov Decision Process (MDP) model to demonstrate the value of not always expelling attackers found in a defender's information system. We have developed models to extract qualitative insights into the interaction of a defender trying to classify an attacker and an attacker trying to evade classification. In particular, we have developed a model in which the attacker chooses an attack rate and the defender chooses a detection threshold to apply after a fixed set of observations. In the model we show that often pure strategy equilibria do not exist. In a related model we allow the defender to dynamically adjust the observation window as he collects data, as in the well known sequential probability ratio test. We show numerically that equilibria appear to exist in the model. In a related model, we restructure the attacker's strategy to be a distribution across the number of hits to try in N steps (a mixed strategy). We show that the equilibrium can be computed efficiently, and we use that fact to extract qualitative insights. One insight is that the defender also ends up using a randomized detection threshold in Nash equilibrium, since with any fixed threshold the attacker will often just attack at a level just below the threshold. This finding suggests that defenders, and hence designers of security software, should consider using randomized detection and classification thresholds. Finally, our methodology allows us to efficiently analyze a broad class of games that are like zero-games except that one player has an extra additive term in their payoff function that only depends on their action. This finding makes a broader class of game models applicable to security settings analyzable.

15. SUBJECT TERMS

computer security, game theory, classification

16. SECURITY CLASSIFICATION OF: N/A			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Musacchio, John
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	SAR	14	**19b. TELEPHONE NUMBER** *(include area code)* +1 510-501-2817

Out-learning Attackers: A Game Theoretic Approach to Cyber Defense

AFOSR Information Operations and Security Program
Final Report
Project Period: 2/1/09 - 11/30/12

John Musacchio, UC Santa Cruz; Greg Frazier and Pat Kreidl BAE Systems

1 Introduction

This project was based on the premise that making systems difficult to infiltrate, detecting and infiltrations as soon as possible, and expelling attackers when detected may not be enough, and in some cases may actually be detrimental. A fundamental problem with this approach is that it encourages attackers to try again, and in subsequent attempts, they are likely to have learned more about the defender's system's defenses than the defender has learned about the attacker. Consequently, each attack is more likely to achieve the attacker's objectives. From this perspective, a cyber defense strategy should not only keep attackers out, but should also enable a defender to learn about an attacker's methods and intentions faster than the attacker can learn about the defender.

Our work during the beginning of the project focused on assessing the potential cost of always expelling attackers versus following an optimized policy that considered the value of keeping attackers in the system to learn about their motives. This work is described in Section 2 of the report. At this stage, the work presumed that the defender could learn about the attacker by observing it in the system, but did not study in detail how that learning would occur. Later, in the project the work focused on this learning – particularly the problem of classifying attackers in situations in which the attacker is adjusting their strategy to make classification more difficult. This part of the investigation is described in Section 3.

2 Expelling Attackers

During the course of an attack, the defender may choose either to expel the attacker once he detects his presence, or to keep his in the system in order to observe and learn about the attacker. If the defender could "out-learn" the attacker, i.e. learn about the attacker faster than he learns about the defender, with the help of that intelligence the defender may be able to totally thwart the attacker's infiltration and ensure the security of the system against this attacker in the long run.

2.1 Model

We use a simple discrete-time MDP to model the system. Our model proceeds in discrete time slots indexed by k. At any time k, the state of the system is described by four state variables. The state variable $c_k \in \{C, NC\}$ describes whether the attacker is "Connected" to the secured information system or "Not Connected". Similarly, $d_k \in \{D, ND\}$ indicates whether the defender has either "Detected" or "Not Detected" the attacker's connection. The other two variables, $x_k, y_k \in [0, 1]$, is used to represent the *knowledge* that the attacker and defender have at time k, respectively.

The system evolves according to the rules described below, and the connection and detection aspects is illustrated in Figure 1.

- Each period that the attacker is disconnected from the defender's system, he may (re-)connect with probability $p_{\text{connect}} = \varepsilon$.
- Each period that the attacker is connected but not yet detected, the defender may detect the existence of attacker with probability $p_{\text{detect}} = \delta$. This probability reflects the capability of the Intrusion

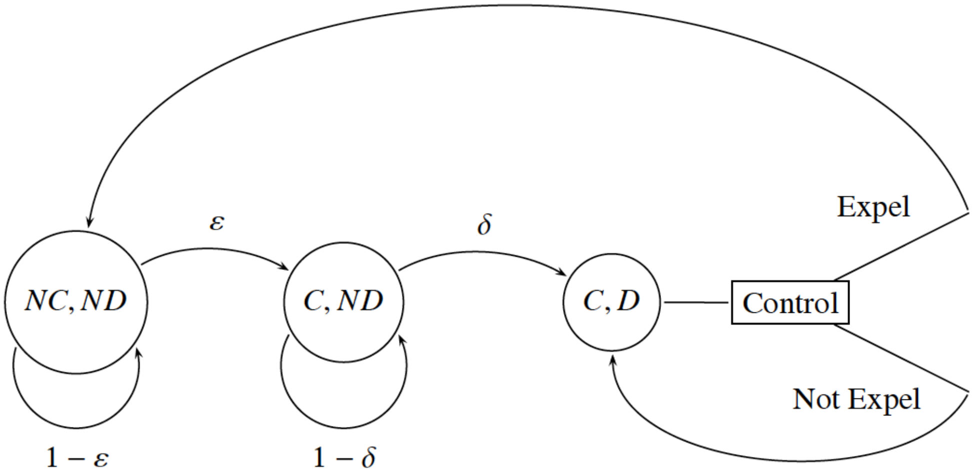


Figure 1: The dynamics of the connection and detection processes.

Detection System (IDS) deployed by the defender.

- After connection (in state $(C,\cdot)$), the attacker could achieve his objective with probability $p_{\text{success}}(k) = \gamma x_k(1-y_k)$ where $\gamma \in (0,1]$ is a scalar. Note that p_{success} increases as the attacker's knowledge x_k grows, and decreases as the defender's knowledge y_k grows. We refer to this formulation as the *single goal* formulation. In an alternative formulation, we consider the case where the attacker may have multiple goals during the course of an attack and each goal will yield 1 unit of reward to the attacker, the quantity $\gamma x_k(1-y_k)$ is then the expected reward for the attacker in each period; and we call it *multiple goals* formulation.

- During periods that the attacker is connected (in state $(C,\cdot)$), he could gather information about the defender's system. The defender may also learn about the attacker during periods that the defender knows the presence (in state (C,D)) of the attacker. To make the model tractable, we use two simple types of learning curves to model the knowledge increase – geometric and linear learning curves. For the *geometric case*, x_k, y_k evolves according to the following recursive expressions during an learning period:

$$\begin{aligned} x_k &= x_{k-1} + \alpha(1-x_{k-1}), \\ y_k &= y_{k-1} + \beta(1-y_{k-1}), \end{aligned}$$

  where $\alpha, \beta \in (0,1)$ are the corresponding learning parameters represents the speed of learning of the attacker and defender, respectively. We also consider the *linear case* which facilitates the analysis of the corresponding MDP problem as to make the state space finite:

$$x_k = \begin{cases} x_{k-1} + \alpha & \text{if } 0 \le k \le \lfloor 1/\alpha \rfloor \\ 1 & \text{otherwise} \end{cases},$$

$$y_k = \begin{cases} y_{k-1} + \beta & \text{if } 0 \le k \le \lfloor 1/\beta \rfloor \\ 1 & \text{otherwise} \end{cases},$$

  where $\alpha, \beta \in (0,1)$ are the slopes of the corresponding learning curves.

- Every period that the attacker is connected and detected, the defender has a control decision to expel the attacker or not. Expelling the attacker will drive the system to state (NC, ND); otherwise, the system stays at (C,D) for one period. This is the only state where the defender has the opportunity to apply control, and the attacker has no control choice in this model.

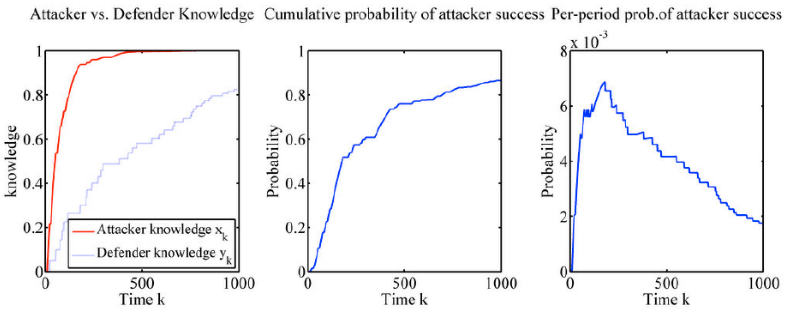


Figure 2: The performance of Always-Expel policy.

2.2 Analysis

We study this model by formulating it as a Markov Decision Process (MDP). The details are given in [1]. One observation that is used in the analysis is that when the defender knowledge y_k reaches 1, the evolution of the model effectively ends since the attacker cannot gain anything more from the system. Thus the model with the linear learning curves should stop evolving after a finite time. This allows us to model it as a Stochastic Shortest Path (SSP) problem [2]. Another observation is that if an attacker has a probability of not returning after an expulsion. it can be modeled as a discounted MDP since future costs are discounted by the chance that the attacker will have given up by that future time. Conversely, a persistent attacker is modeled with an undercounted MDP. For the persistent attacker, we can use these observations to show some monotonicity properties of the defender's value function which in turn lead to this result

Prop. 1 *For the undiscounted MDP (persistent attacker) with linear learning curves, the* Never-Expel *policy is optimal and dominates any other policy.*

Some further technical arguments extend the above proposition to the other, geometric learning curve as well. We also consider the following embellishment to the original model.

- *Boosting Factor Upon Expulsion:* A simple embellishment of the original model is to introduce a knowledge boosting factor, f. When the defender chooses to expel the attacker, the attacker's knowledge grows according to $x_k = x_{k-1} + f\alpha(1 - x_{k-1})$ (geometric case) or $x_k = x_{k-1} + f\alpha$ (linear case) where $f > 1$ and k is the time index. This expression reflects the possibility that the attacker may learn faster in an expulsion period than in a period he stays connected without expulsion. This reflects the effect that the attacker may learn something about the reason of his failure (being detected) so that he can improve tactics next time. This embellishment shall not affect the results derived in this section because it only makes expulsion less attractive.

2.3 Simulation Results

Without formulating the model into an MDP, one can simulate the evolution of the attack-defense process under different defender policies. To begin with, it is interesting to compare two extreme policies Always-Expel and Never-Expel. In Figure 2, 3 the result of a typical sample path is displayed for the geometric case. The performance measure is defined as the cumulative probability of attacker success (the single goal formulation), and the parameter choice is $\alpha = .02, \beta = .05, \gamma = .01, \varepsilon = .05, \delta = .05$. Besides, we assume at the initial state ($k = 0$) both attacker and defender's have zero knowledge and the attacker is not connected (state (NC, ND)), and simulate the system from time $k = 0$ to 1000. The left plot of Figure 2 shows that the attacker's knowledge grows faster than that of the defender under the Always-Expel policy, and the

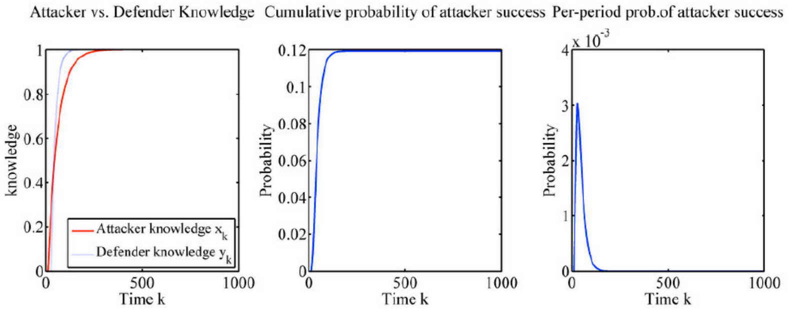
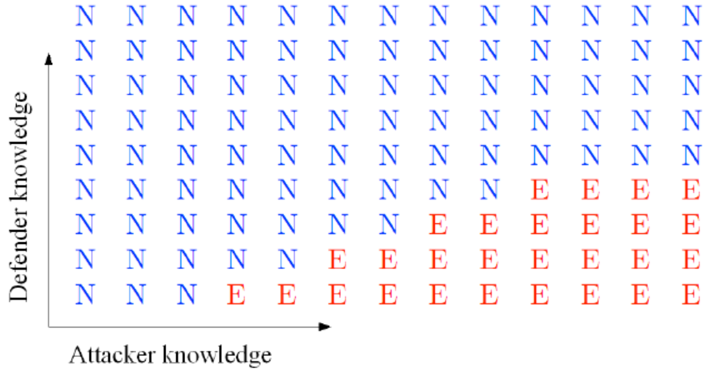


Figure 3: The performance of Never-Expel policy.

middle plot indicates that the cumulative probability of attacker success exceeds 90%. Similar plots are provided in Figure 3 for the Never-Expel policy. By comparing the left plots of Figure 2 and 3, one could see that the defender's knowledge grows faster in the latter and the defender successfully *out-learn* the attacker. Consequently, the cumulative probability of attacker success is under 12%, an order of magnitude of improvement from the rather naive Always-Expel policy. The right plots in both figures demonstrate the evolution of the per-period probability of attacker success. One could observe the rapid defender learning under the Never-Expel policy results in the drastic drop of the per-period attacker success probability which further explains the advantage of the out-learning strategy.

2.3.1 Structure of the Optimal Policy

By way of policy iteration, it is easy to compute the optimal stationary policy consisting of decisions only depends on states. Figure 4 illustrates the optimal policy of the discounted MDP under the parameter choice: $\rho = .89, \alpha = .09, \beta = .13, \gamma = .05, \varepsilon = .05, \delta = .05$. The optimal policy is displayed in a control-matrix form where each entry corresponds to the optimal decision of the defender in state $(x, y, \mathbf{3})$. By experiments with various parameter settings, we observe that the optimal policy of the discounted MDP always possesses a lower-triangular, threshold-like structure. Roughly, with a fixed amount of knowledge the defender shall switch from Not-Expel to Expel as the attacker knowledge grows. Note this is similar to the idea of a *threshold policy* where the optimal control switch from one to another as the state exceeds some threshold point. This is quite intuitive in that for fixed defender knowledge, the more the attacker knows about the defender's system, the more immediate damage he may impose. Consequently, it might be more preferable to expel the attacker and avoid relatively significant immediate costs for a while; moreover,


Defender knowledge
Attacker knowledge

N	N	N	N	N	N	N	N	N	N	N	N	N
N	N	N	N	N	N	N	N	N	N	N	N	N
N	N	N	N	N	N	N	N	N	N	N	N	N
N	N	N	N	N	N	N	N	N	N	N	N	N
N	N	N	N	N	N	N	N	N	N	N	N	N
N	N	N	N	N	N	N	N	N	E	E	E	E
N	N	N	N	N	N	N	E	E	E	E	E	E
N	N	N	N	N	E	E	E	E	E	E	E	E
N	N	N	E	E	E	E	E	E	E	E	E	E

N : Not Expel, E: Expel

Figure 4: The optimal policy in control-matrix form.

since there is some positive probability that the attacker may give up in each period, postponing the attack by expulsion is more advantageous than the out-learning strategy. On the other hand, as the defender knowledge grows larger there are less Expel entries in the optimal control-matrix. Again, this could be understood from the fact that the immediate cost is decreasing w.r.t the defender knowledge.

2.4 Conclusion

We have considered the defender's policy optimization problem with presence of the learning effect in a security context. By formulating the problem into a Markov decision process, we are able to analyze the characteristics of the optimal policy. If the attacker's is persistent, the optimal strategy of the defender is to keep the attacker in the system in order to out-learn him and eventually thwart the attacks, which is quite different from the conventional idea of expelling the attacker whenever detecting his presence. For the case where the attacker may give up, it can be formulated into a discounted MDP, and we observe that the optimal policy in this case has certain structure by numerical experiments.

Our model, although quite stylized, is able to capture the interesting effects when one considers the learning effect in a cyber-defense scenario. It demonstrates the potential benefit of gathering intelligence from the attacker during the course of a defense. This idea yields a new perspective in studying the network security problems.

3 Classification

Our work on classification can be divided into two categories. Our work earlier in the project looked at a model in which the attacker chooses a single number, the rate at which to attack. This work is described in Section 3.1. The second category of work, done later in the project, supposed the attacker could pick a distribution across "attack strengths" (mixed strategy). This work is described in Section 3.2.

3.1 Attacker Chooses Real Number Valued "Attack Strength"

This section summarizes work published in [3] The model is illustrated in Figure 5. A network defender faces an attacker that can either be a spy or spammer with probabilities p and $1-p$ respectively. The defender has two servers that can be attacked, a File Server (FS) and a Mail Server (MS). We suppose that spammers attack the MS most often because they want to send spam and to get the addresses of potential victims. However, a spammer occasionally hits the FS as he explores the defender's information system looking for other potential targets. We suppose time is discrete, and in each period k, a spammer hits the FS with probability $\theta_0 < \frac{1}{2}$ and otherwise he hits the MS. The attacks are restricted to be i.i.d. Bernoulli from period to period. Moreover, we suppose the defender can observe the sequence of attacks $z_k \in \{MS, FS\}$. Spammers are supposed to be non-strategic, so θ_0 is taken to be a fixed parameter in the model.

A spy has to choose the frequency with which to hit the FS, which is what he prefers to attack as that is where the information he wants is stored. However, he also can choose to hit the MS during some time periods to make it more difficult for the defender to distinguish him from a spammer. We suppose that the spy's strategy is to pick a single probability θ_1 of hitting the FS in any period. Once the spy picks θ_1, his attacks on the FS are restricted to be Bernoulli. If he picks θ_1 too high, then it will be easy for the defender to distinguish him from a spammer; if he picks θ_1 too low, then he reduces the frequency with which he gets to attack the desired target.

The defender has to decide in each period whether to classify the attacker as a spy or a spammer, or to do nothing and keep observing. When a spammer is attacking, the defender pays a penalty c_0 each time he hits the MS and pays a penalty F for mis-classifying a spammer as a spy. When a spy is attacking,

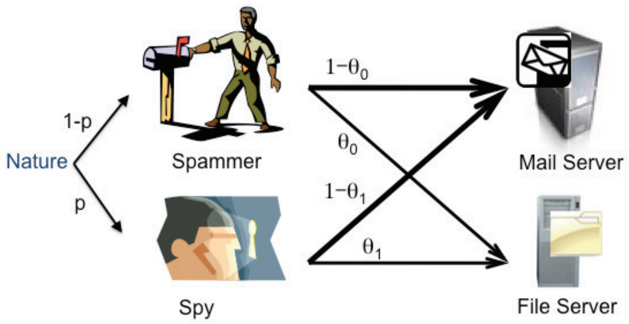


Figure 5: An illustration of the classification game.

the defender pays a penalty c_1 for each hit on the FS, which appears as a payoff $-c_1$ to the spy. If the defender correctly classifies a spy, the game ends and the spy pays a penalty L. However, if the defender mis-classifies a spy as a spammer, we suppose that the spy can then continue to attack with impunity and thus earns a reward equal to the discounted net present value of an endless stream of FS attacks that happen with probability θ_1 in each period. This mis-classification reward to the spy, like the the spy's rewards for all preceding FS attacks, appears as a penalty to the defender.

In this work we consider two versions of this game. In the first version (Section 3.1.1), we suppose that the defender's strategy is to commit to a fixed number of observation periods N. Simultaneously, spies pick θ_1. At the end of N periods, the defender makes the classification decision that minimizes his expected cost given the observations. We call this the fixed N game. We find that this game has no pure Nash equilibrium by experiments covering the whole parameter space.

In the second version (Section 3.1.2), the defender does not commit to an observation period but instead can decide to keep taking observations, depending on what has been observed so far. We call this the dynamic N game. The defender's best response to a given θ_1 is similar to the well known Sequential Probability Ratio Test (SPRT) [4]. In a SPRT (with Binomial data and two hypothesis), one keeps track of the Log Likelihood Ratio (LLR), which evolves like a one-dimensional random walk as observations come, and makes a classification when the LLR crosses either an upper or lower threshold. For a given θ_1, the best response of the defender is to use an SPRT-like test with particular thresholds (that can be numerically computed) and a hypothesis for θ_1 that matches the value the spy is actually using. If we fix a defender strategy (SPRT thresholds and hypothesis $\hat{\theta}_1$) it might be that the spy's best response is to play with a θ_1 that does *not* match what the defender is expecting. A Nash equilibrium of the game would be a point where attacker θ_1 and the defender's hypothesis $\hat{\theta}_1$ match. We find that the dynamic N game can exhibit such a Nash equilibrium by experiments leveraging available computational tools for Partially-Observable Markov Decision Processes (POMDPs) [5] and other finite-state Markov reward processes [2].

3.1.1 Fixed N game

In the Fixed N game, the defender employs a fixed-sample-size, uniformly most powerful (UMP) test [6]. We start by considering a simple-vs-simple test $H_0 : \theta = \theta_0$ versus $H_1 : \theta = \theta_1$. Recall that the observations of server hits are modeled as a sequence of i.i.d Bernoulli random variables conditioned on the true type of an attacker. Therefore, the likelihood-ratio, given a vector of observations $\mathbf{z}_N = (z_1, \ldots, z_N)$, is given by $\lambda(\mathbf{z}_N) = \mathbf{P}[\mathbf{z}_N \mid X = 1] / \mathbf{P}[\mathbf{z}_N \mid X = 0] = [\theta_1(1-\theta_0)/(1-\theta_1)\theta_0]^z \cdot [(1-\theta_1)/(1-\theta_0)]^N$ where $z := \sum_{k=0}^{N} z_k$ simply counts the number of FS attacks. By the Neyman-Pearson lemma [6], a test with decision rule "rejecting H_0 if $\lambda(\mathbf{z}_N) > M$" and a Type-I error probability α is a level α UMP test; moreover, it is easy to check the condition $\lambda(\mathbf{z}_N) > M$ is equivalent to $z > m$ for some integer m. However, the defender has no access to the value of θ_1 chosen by a spy in our game since both players act simultaneously. Therefore, the defender in fact carries out an one-sided test $H_0 : \theta \leq \theta_0$ vs $H_1 : \theta > \theta_0$. By properly choosing an alternative hypothesis $H_1' : \theta = \hat{\theta}_1$, the defender may effectively transform the one-sided test into a simple-vs-simple one; moreover, by Karlin-Rubin theorem [6] the aforementioned decision rule still yields a test with the smallest mis-detection rate among all tests with the desired false-alarm rate level.

From above discussion, we see that the defender's strategy is a pair of non-negative integers (N, m) such that $m < N$.

To study the existence of pure Nash equilibrium, we adopt the standard approach of deriving best response mappings of both players and checking for intersections point(s). Due to the complexity of our payoff functions, there are no close-form expressions for the best responses and we resort to extensive numerical experiments. It turns out that a pure Nash equilibrium fails to exist in our Fixed N game. For many problem instances, we find that the the attacker's best response function is discontinuous. For these examples, When a defender commits to a large N, the attacker's best response is to attack aggressively by choosing a θ_1 that's large. Conversely, when the defender commits to a small N, the attacker's best response is the pick a θ_1 close to θ_0 to avoid detection. This discontinuity makes it such that the best response functions of the two players never intersect – and such an intersection is what is needed to have a Nash equilibrium point.

3.1.2 Dynamic N Game

In this section, we remove the restriction that the defender commits to a fixed observation time. That is, as in the famous Wald problem [4], the number of observations N before classification depends not just on the two players' strategies but also on the particular observation sequence $\mathbf{z}_k = (z_0, z_1, \ldots, z_k)$. The spy's problem remains essentially the same as in the preceding section, namely to select how frequently to hit the file-server relative to the mail-server (i.e., the value of probability θ_1). Note that the spy has no obligation (and, in fact, generally has incentives not) to behave as hypothesized by the defender (i.e., the spy's parameter θ_1 may differ from the value $\hat{\theta}_1$ hypothesized by the defender). The question we seek to answer is whether it is possible for the defender to hypothesize a value for $\hat{\theta}_1$, and design his best response strategy accordingly, such that the spy's best response yields $\theta_1 = \hat{\theta}_1$.

If the defender were to hypothesize the true value of θ_1, then results for the Wald problem imply that the defender's best response function takes the form of a Sequential Probability Ratio Test (SPRT) parametrized by two probability thresholds we denote by η and $\xi > 1 - \eta$ i.e., initialize probability $b_{-1} = p$ and, in each stage $k = 0, 1, 2 \ldots$, first apply the probabilistic state recursion

$$\mathbf{P}\left[X = 1 \mid \mathbf{z}_k\right] \equiv b_k = \begin{cases} \dfrac{(1-\theta_1)b_{k-1}}{(1-\theta_0)(1-b_{k-1}) + (1-\theta_1)b_{k-1}} & , \text{ if } z_k = \text{MS} \\[2ex] \dfrac{\theta_1 b_{k-1}}{\theta_0(1-b_{k-1}) + \theta_1 b_{k-1}} & , \text{ if } z_k = \text{FS} \end{cases}$$

and then choose to `classify-spammer` if $b_k < 1-\eta$, to `classify-spy` if $b_k > \xi$, and to `continue` otherwise. Under the assumption that the defender possesses no knowledge on how the spy may play, the only option is to employ the SPRT strategy for some hypothesis $\hat{\theta}_1$ on the spy's strategy. We denote such hypothesis-dependent SPRT thresholds by $\eta(\hat{\theta}_1)$ and $\xi(\hat{\theta}_1)$. We similarly denote the defender's associated cost in by $J^D(\hat{\theta}_1|\theta_1)$, also reflecting its dependence on the spy's true choice of θ_1. Recognizing the defender's best response model as a special case of the well-studied Partially Observable Markov Decision Process (POMDP) [5], we appeal to a publicly available POMDP solver (see http://www.pomdp.org) to both optimize the SPRT thresholds and compute the defender's performance $J^D(\hat{\theta}_1|\hat{\theta}_1)$ if the hypothesis were in fact true.

The spy's best response considers the defender's strategy as given, i.e., hypothesis $\hat{\theta}_1$ and the associated SPRT thresholds $\eta(\hat{\theta}_1)$ and $\xi(\hat{\theta}_1)$ are known to the spy. In turn, for any choice of the true θ_1, denote the spy's associated cost by $J^A(\theta_1|\hat{\theta}_1)$. It follows that the spy's best response is the value of $\theta_1 \in (\theta_0, 1]$ that minimizes $J^A(\theta_1|\hat{\theta}_1)$, or equivalently the value of θ_1 that maximizes the incentive $J^A(\hat{\theta}_1|\hat{\theta}_1) - J^A(\theta_1|\hat{\theta}_1)$ to deviate from the defender's hypothesis $\hat{\theta}_1$. Our computation of the spy's response relies on a nonlinear program, each iteration on a candidate value for θ_1 involving the construction and solution of a finite-state Markov chain that exploits two properties of the defender's SPRT strategy. Firstly, the defender's probabilistic state recursion can (until classification) be equated with a random walk along the real line involving the defender's log-likelihood ratio (LLR)

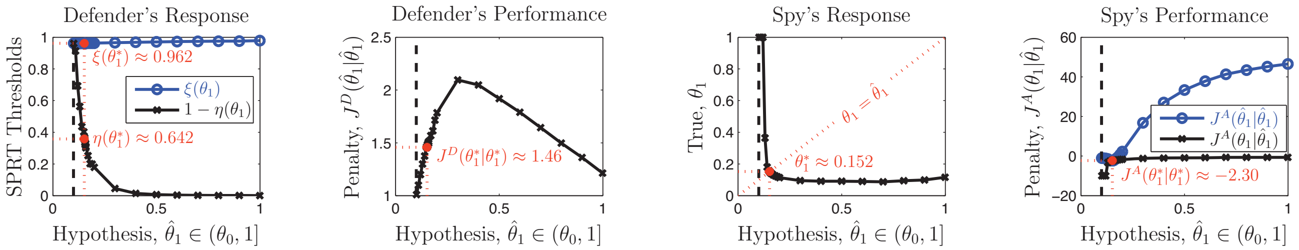
$$R_k = \log\left(\frac{\mathbf{P}\left[\mathbf{z}_k \mid X=1\right]}{\mathbf{P}\left[\mathbf{z}_k \mid X=0\right]}\right) = \begin{cases} R_{k-1} + \log\left(\frac{1-\hat{\theta}_1}{1-\theta_0}\right) & , \text{ if } z_k = \text{MS} \\ R_{k-1} + \log\left(\frac{\hat{\theta}_1}{\theta_0}\right) & , \text{ if } z_k = \text{FS} \end{cases},$$

starting from the origin $R_{-1} = 0$. In turn, the SPRT thresholds (and prior probability p) determine the segments of the real-line corresponding to the three control actions available to the defender i.e., choose to `classify-spammer` if $R_k < \log\left(\frac{(1-p)\left[1-\eta(\hat{\theta}_1)\right]}{p\eta(\hat{\theta}_1)}\right)$, to `classify-spy` if $R_k > \log\left(\frac{(1-p)\xi(\hat{\theta}_1)}{p\left[1-\xi(\hat{\theta}_1)\right]}\right)$, and to `continue` otherwise.

Secondly, the the spy's strategy θ_1 alters the statistics of this random walk, lower (higher) values increasing the chances that the LLR first exits the continue region at the lower (upper) end of the real-line. The Markov chain representation involves $Q+3$ states, Q of them indexing the levels of a uniform quantization of the LLR continue region, one indexing an initial state and two indexing terminal states (one per classify decision). The transition probabilities reflect not only the spy's response θ_1 and the increments $R_k - R_{k-1}$ of the defender's LLR walk, but also the noise introduced by the quantization. The transition costs reflect the spy's rewards from file-server attacks and evading detection, as well as the spy's cost of actual detection. Then, in each iteration of the nonlinear program, standard techniques for Markov chains [2] can be employed to approximate the expected total discounted cost when θ_1 is not necessarily equal to $\hat{\theta}_1$. We omit further details here, but the outputs of this method are a particular value for the spy's policy parameter $\theta_1 \in (\theta_0, 1]$ and the spy's associated cost $J^A(\theta_1|\hat{\theta}_1)$.

For the game-aware defender, the key question is whether there exists a hypothesis $\hat{\theta}_1$ from which the spy has no incentive to deviate, choosing $\theta_1 = \hat{\theta}_1$. Fig. 6 illustrates an empirical solution obtained via the computational methods and approximations discussed above, where for each hypothesis $\hat{\theta}_1$ we

1. employ the POMDP solver to obtain for the defender's (a) SPRT thresholds $\eta(\hat{\theta}_1)$ and $\xi(\hat{\theta}_1)$ as well as (b) penalty $J^D(\hat{\theta}_1|\hat{\theta})$; then



(a) (b) (c) (d)

Figure 6: Equilibrium solution to the dynamic N game with $p = 0.5$, $\theta_0 = 0.1$, $\delta = 0.95$, $c_0 = 0.01$, $c_1 = 1$ and $L = F = 50$. Each marked point on the defender's response and performance curves (plots (a) and (b), respectively) is obtained via the POMDP solver, while each marked point on the spy's response and performance curves (plots (c) and (d), respectively) is obtained via the nonlinear program iterating on a quantized approximation (with $Q = 100$) of the defender's SPRT solution. The equilibrium point is where the spy's response curve in (c) intersects the $\theta_1 = \hat{\theta}_1$ line.

2. employ the nonlinear program to obtain the spy's (c) true θ_1 and (d) the penalty $J^A(\theta_1^A|\hat{\theta}_1)$ (where we also show its comparison to penalty $J^A(\hat{\theta}_1|\hat{\theta}_1)$ were the spy to behave as the defender hypothesizes).

Our procedure starts with values for $\hat{\theta}_1$ over the whole interval $(\theta_0, 1]$ in increments of 0.1. Then, it identifies each sub-interval in which, under the assumption of continuity, there may lie a point in which $\theta_1 = \hat{\theta}_1$. The procedure continues with values for $\hat{\theta}_1$ over each such sub-interval in increments of 0.01, and so on. The procedure terminates at a pre-specified precision, which in Fig. 6 was set to 0.001. The markers on each curve in Fig. 6 denote the selected values of $\hat{\theta}_1$ over the entire procedure. The solution point in each plot corresponds to the value of $\hat{\theta}_1$ at which the spy's response θ_1 in Fig. 6(c) is nearest to the $\theta_1 = \hat{\theta}_1$ line.

Observe that the two players' response functions exhibit a smooth confusion versus exploitation trade-off. That is, for hypotheses $\hat{\theta}_1$ close to θ_0, the defender's SPRT thresholds are such that spammer classifications are made frequently and nearly immediately: in other words, when the defender anticipates a spy favoring confusion, his strategy reduces to near-immediate expulsion of either type of attacker and, in turn, the spy's response is to hit the FS at every opportunity. For hypotheses $\hat{\theta}_1$ away from θ_0, the defender's SPRT thresholds are such that classification is deferred: in other words, when the defender anticipates a spy favoring exploitation, his strategy allows for the time to reliably classify either type of attacker and, in turn, the spy's response is to evade detection by hitting the FS almost as infrequently as spammers do. For the model parameters in Fig. 6, the equilibrium point of $\theta_1^* \approx 0.152$ neutralizes all incentive for the spy to either confuse an exploitation-oriented defense or to exploit a confusion-oriented defense.

3.1.3 Conclusion

In this part of the project, we developed a security classification game. The defender tries to effectively classify the attackers (spammer or spy) while controlling the damage during the period of gathering evidence. A strategic spy faces the trade-off between (i) exploiting the defender's observation time by attacking aggressively and (ii) confusing the defender by mixing attacks thereby enjoying the benefits of mis-detection. The non-existence of pure Nash equilibrium of our fixed N game suggests that an over-simplified strategy adopted by the defender will never lead the game to settle to a stable point where both players behave predictably. This problem is mitigated by allowing the defender to make decisions at each period of time in our dynamic N game, which essentially dis-incentivizes the spy's response to drastically shift from aggressive exploitation to moderate confusion.

3.2 Attacker Chooses Mixing Distribution

The results of this section are described in more detail in [7].

3.2.1 Basic Model

The game model is as follows. As in the earlier family of models, Nature decides the type of an attacker in a network: spy or spammer with probabilities p and $1 - p$ respectively. The network consists of a defender and two servers that might be attacked: a File Server (FS) with sensitive data and a Mail Server (MS) with contents of inferior importance. The spy's goal is to attack the FS as frequently as possible while evading detection, and the spammer's goal is to attack the MS to congest the network or annoy the defender. The defender is a strategic player who monitors the two types of servers at each time slot (we consider discrete time). We assume a constant classification window of N time slots, during which the defender observes the number of attacks to the FS. The spammer is a non-strategic player, who attacks on the FS S time slots with a known cumulative distribution function. For instance, he can be modeled to have a Bernoulli distribution at each time slot with a small per-period probability θ_0 of a hit on the FS. For a fixed observation window of N time slots, the defender selects the threshold T of time slots, below which he classifies the attacker as a spammer and the spy selects the number of FS attacks H to launch.

Attacker's cost function: The spy is detected when the defender's threshold T is smaller or equal to the spy's selection of H (the number of FS attacks). In this case, he has a cost of c_d. We assume that each FS hit gives the spy some benefit captured by the parameter c_a. We also assume that he gains nothing from attacking the MS. His overall gain from the attacks is proportional to the number of time slots H he selected to attack. Since it will be useful to work with a cost function for the attacker rather than a payoff function, we subtract the gain from the attacks. Thus, his overall cost function can be expressed as follows

$$J_A(T,H) = c_d \cdot \mathbf{1}_{T\leq H} - c_a \cdot H,$$

where $\mathbf{1}_{T\leq H}$ is 1 if $T \leq H$ and 0 otherwise.

Defender's reward function: The defender's expected reward function depends on the true type of the attacker. In the case that he faces a spy (which happens with probability p), he makes a correct classification and gains c_d when his threshold $T \leq H$. He always gets a cost from the FS attacks which is proportional to H. With probability $1-p$ he faces a spammer who selects to attack S time slots. For a fixed T, we denote by $\phi(T) = \Pr\{S \geq T\}$ the probability that the spammer attacks at least T times on the FS. Then, the defender has an expected false alarm penalty of $c_{fa} \cdot \phi(T)$ and his total expected payoff is

$$\tilde{U}_D(T,H) = p \cdot (c_d \cdot \mathbf{1}_{T\leq H} - c_a \cdot H) - (1-p) \cdot c_{fa} \cdot \phi(T).$$

By scaling the above function, we finally get

$$U_D(T,H) = c_d \cdot \mathbf{1}_{T\leq H} - c_a \cdot H - \mu(T),$$

where $\mu(T) = \frac{1-p}{p} \cdot c_{fa} \cdot \phi(T)$. We assume that $\phi(T)$ is strictly decreasing with T.

3.2.2 Players' interactions

For a fixed classification window N the spy has $N+1$ available actions: attack the file server $H \in \{0,\dots,N\}$ times, whereas the defender has $N+2$ available actions: select $T \in \{0,\dots,N+1\}$ as the classification threshold. A threshold of 0 always results in spy classification (as any intruder will attack the FS at least 0 times), and a threshold of $N+1$ always results in spammer classification.

We model our problem as a nonzero-sum game, where the term in the defender's payoff that is different than the spy's cost depends only on the defender's strategy. In the literature these games are known as almost zero-sum games or quasi zero-sum games. We are interested in Nash equilibria in mixed strategies for the following reason. On the one side, the spy seeks to select a number of attacks just below the defender's threshold. On the other side, the defender aims to select a threshold equal to the attacker's strategy. Thus the players need to mix between different strategies to make themselves less predictable. The spy chooses a distribution $\boldsymbol{\alpha}$ on the available numbers of FS hits – thus $\boldsymbol{\alpha}$ is a vector of size $N+1$ with non negative elements that sum to 1. Similarly the defender chooses a distribution $\boldsymbol{\beta}$ on the collection of possible thresholds T. Thus $\boldsymbol{\beta}$ is a vector of size $N+2$.

3.2.3 Game-Theoretic Analysis

In this section, we state our main theorem. We use the notation "min" when we find the minimum element of a vector, and "minimize" when we minimize a specific expression over some constraints. We use the superscript T for matrix transposition.

Let Λ be a $(N+1)\times(N+2)$ matrix representing the spy's strategies' cost for any possible strategy of the defender. We shift Λ by a constant parameter $Nc_a + \epsilon$, with $\epsilon > 0$. Thus, Λ can be written in the following form

$$\tilde{\Lambda} = c_d \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 1 & 1 & \dots & 0 & 0 \\ 1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \dots & 1 & 0 \end{pmatrix} - c_a \begin{pmatrix} 0 & 0 & \dots & 0 \\ 1 & 1 & \dots & 1 \\ 2 & 2 & \dots & 2 \\ \vdots & \vdots & \vdots & \vdots \\ N & N & \dots & N \end{pmatrix}$$

The last all-zero column in the first component of $\tilde{\Lambda}$ captures that is never caught when the defender chooses the $N+1$ threshold. With $\tilde{\Lambda}$ defined as above, the attacker cost can be written as $\boldsymbol{\alpha}^T\tilde{\Lambda}\boldsymbol{\beta}$ and the defender payoff can be written as $\boldsymbol{\alpha}^T\tilde{\Lambda}\boldsymbol{\beta} - \boldsymbol{\mu}^T\boldsymbol{\beta}$. It will turn out that certain computations are simplified by using a a matrix with only positive entries. We therefore define

$$\Lambda = \tilde{\Lambda} + (Nc_a + \epsilon) \cdot \mathbf{1}_{(N+1)\times(N+2)}$$

where $1_{(N+1)\times(N+2)}$ is a matrix of all ones of dimension $(N+1)\times(N+2)$. Since $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ must each sum to 1, the expressions $\boldsymbol{\alpha}^T\Lambda\boldsymbol{\beta}$ and $\boldsymbol{\alpha}^T\Lambda\boldsymbol{\beta} - \boldsymbol{\mu}^T\boldsymbol{\beta}$ are respectively the attacker cost and defender payoff shifted by a constant. Since adding a constant to a players payoff does not affect their best responses, from here on we will consider these expressions to be the payoff functions of each player.

For a given defender strategy, $\boldsymbol{\beta}$, the minimum attacker cost is achieved by putting positive probability only on strategies corresponding to the minimum entries of the vector $\Lambda\boldsymbol{\beta}$. Such a strategy results in a attacker cost of $\min[\Lambda\boldsymbol{\beta}]$ where min extracts the minimum element of the vector. The defender's payoff when the attacker plays a best response is

$$\theta(\boldsymbol{\beta}) = \min[\Lambda\boldsymbol{\beta}] - \boldsymbol{\mu}^T\boldsymbol{\beta}.$$

This function is important for our subsequent analysis. Since it is a measure of how "good" a strategy $\boldsymbol{\beta}$ is, we refer to $\theta(\boldsymbol{\beta})$ as the *defendability* of $\boldsymbol{\beta}$. This is similar to the concept of "vulnerbaility" developed in [8].

Lemma 1 *In NE, the defender strategy $\boldsymbol{\beta}$ must maximize $\theta(\beta)$.*

Proof 1 (Proof Sketch) *The minimum cost the attacker can achieve in response to $\boldsymbol{\beta}$ is $\delta := \min[\Lambda\boldsymbol{\beta}]$. In Nash Equilibrium, the attacker must be playing a best response and the defender must not be able to improve payoff with a unilateral deviation. The attacker's optimization problem, subject to the constraint that he pick a strategy that makes the defender unable to improve payoff from a unilateral deviation takes the form*

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\textit{minimize}} \quad & \boldsymbol{\beta}^T\Lambda^T\boldsymbol{\alpha} \\ \textit{subject to} \quad & \boldsymbol{\alpha} \geq \mathbf{0}, \mathbf{1}^T\boldsymbol{\alpha} \geq 1, \\ & \Lambda^T\boldsymbol{\alpha} - \boldsymbol{\mu} \leq \theta(\boldsymbol{\beta})\mathbf{1}. \end{aligned}$$

The solution of this problem needs to be δ, since if it were more than δ, the attacker would not be achieving the minimum possible cost. However, analysis of the dual of this program shows that the problem yields a solution of δ only if β is a maximizer of the function $\theta(\beta)$. The details of the dual program analysis are left out here for space constraints.

In NE the defender maximizes defendability, or equivalently he picks a solution of the following LP:

$$\begin{aligned} \underset{\boldsymbol{\beta},z}{\text{maximize}} \quad & -\boldsymbol{\mu}^T\boldsymbol{\beta} + z \\ \text{subject to} \quad & z\mathbf{1} \leq \Lambda\boldsymbol{\beta} \\ & \mathbf{1}^{\boldsymbol{T}}\boldsymbol{\beta} = 1. \end{aligned} \tag{1}$$

Table 1: Defender's strategy in NE ($\beta_m = c_a/c_d$)

#	...	β_s	β_{s+1}	...	β_N	β_{N+1}
1.	0	0	β_m	β_m	β_m	$1-(N-s)\beta_m$
2.	0	$1-(N-s)\beta_m$	β_m	β_m	β_m	0

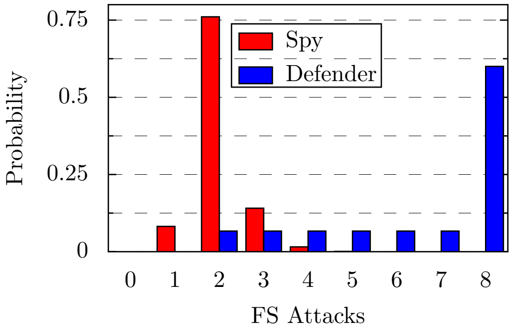


Figure 7: *Players's best responses in NE for $N = 7,\ \theta_0 = 0.1, c_d = 15,\ c_a = 1,\ c_{fa} = 23,\ p = 0.2$.*

As we can see from the LP the defendability is maximized at one of the extreme points of the polyhedron defined by $\Lambda x \geq 1$. Given an extreme point $\boldsymbol{x}$, the corresponding distribution is $\boldsymbol{\beta} = \dfrac{\boldsymbol{x}}{\|\boldsymbol{x}\|}$.

Theorem 1 *In any Nash equilibrium the defender's strategy $\boldsymbol{\beta}$ maximizes the defendability. A maximizing value of $\boldsymbol{\beta}$ exists amongst one of the two forms in Table 1 for some s. If there is only one maximizing $\boldsymbol{\beta}$ amongst vectors of the form in Table 1, then the Nash equilibrium is unique.*

The theorem is shown by showing that an extreme point vector that corresponds to a maximizing distribution vector of defendability has certain properties. Most importantly, there needs to be one contiguous block of tight inequalities in the equations $\Lambda x \geq 1$. Using that fact, one can show that if s and f are the start and finish indices of the contiguous block, then β_{s+1} through β_f needs to equal c_a/c_d. Other properties can be used to show that f must either be N or $N+1$.

3.2.4 Numerical Results/Simulations

We conducted various experiments for different sets of parameters N, c_a, c_d and p, assuming that the spammer attacks with Bernoulli distribution with parameter θ_0. We first used the methods discussed above to calculate the strategies of both players at equilibrium. We later used the Gambit software [9] and validated our theoretical results. We present here two characteristic examples, to illustrate the two possible structures of the Nash equilibria (the two aforementioned cases).

Figure 7 illustrates Case 1 and the unique Nash equilibrium for $N = 7$ time slots. As we can see, all the middle points are given the same weight $x_m = c_a/c_d = 0.0667$, $x_s = 0$ and $x_{f+1} > x_m$. The structure of the equilibria is given by the first row of Table 1, with $s = 1$ and $f = 7$.

Figure 8 presents the unique Nash equilibrium for $N = 7$ in Case 2. As we can see, again all the middle points are given the same weight $x_m = c_a/c_d = 0.1$, but here $x_s > x_m$ and $x_{f+1} = 0$. Note that as p increases, larger weight is given to the smallest threshold, in order to detect the most-probable-to-exist spy. We also observe that the defender still gives some weight on the larger thresholds and is not focused on a range around $N\theta_0$. This can be explained from the strictly decreasing false alarm cost function μ: the defender has always an incentive to use larger thresholds to increase his expected payoff.

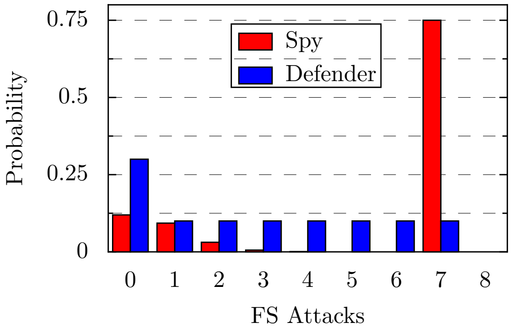
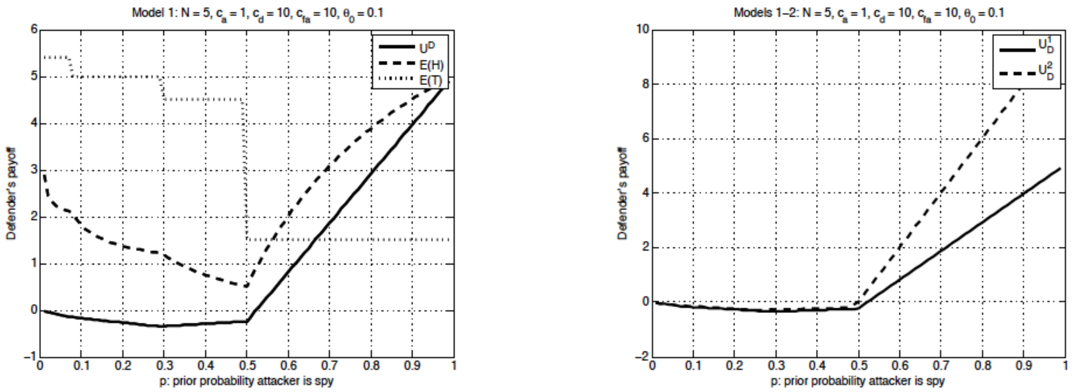


Figure 8: *Players's best responses in NE for* $N = 7,\ \theta_0 = 0.1, c_d = 10,\ c_a = 1,\ c_{fa} = 10,\ p = 0.8.$



(a) NE defender payoff first decreases and then increases on p

(b) Comparison of models 1-2

Figure 9: Numerical results of revised model with forensics.

3.2.5 Model Extension to Study Value of Forensics

We have extended this model to study a situation in which if the spy is detected, the cost to him is proportional to how hard he attacked. The idea here is that if he attacked harder, the defender will have more evidence to analyze to learn about the attacker. By comparing the equilibrium payoffs with and without this feature, one can get a measure of the value to an organization of investing in forensics capabilities (since without theses capabilities one could not use the evidence left by the attacker against him). This study is detailed in our paper [10]. This work also generalizes the results of this spy-defender game to apply to more general payoff functions than those described here. Figure 9 illustrates some numerical results from this study. The first panel shows the expected value of threshold and attach strength H, and the equilibrium defender payoff for the revised model, as a function of the prior probability p that the attacker is a spy. The second panel compares the defender payoff in the revised model (with forensics) to the older model (without).

3.2.6 Quasi Zero-Sum Games

In this work, the game model has the feature that it looks “almost” like a zero-sum game. The defender's payoff is the opposite of the spy's, plus an extra term than only depends on the defender's action. This structure we call a “quasi zero-sum game.” Our work on this model shows that equilibria of quasi zero-sum games can be found by solving a Linear Program (LP), just as true zero-sum games are widely known to be solvable with an LP. This finding is important because quasi zero-sum games can potentially model a wide range of practical problems in the information security domain. We are currently preparing a paper discussing this finding and planning to submit it to an operations research journal.

4 Conclusions and Impact

In this project we have demonstrated the potential value of not always expelling attackers detected in the system. This important observation has potential impact in real-world applications. We have also studied the problem of classifying attackers that are trying to evade classification. The closed-loop behavior of attackers trying to remain below a detection threshold results in Nash Equilibria being mixed in most situations. This qualitative finding suggests that developers of security software doing classification or intrusion detection should consider using randomized thresholds. The same observation may apply to spam filtering software. The qualitative findings from this work, we hope, will have an impact on developers of such systems. However, there are many unanswered questions that these findings lead to, such as how can a designer of security software choose the right randomization strategy? Questions like this are likely to be a topic of future research for us, and perhaps others in the research community.

Our results on quasi zero-sum games also have a great deal of potential impact. There are a large number of practical situations that are modeled much more accurately by a quasi zero-sum game than a true zero-sum game. The ability to efficiently find equilibria of this broader class of games may have great potential impact by enabling researchers to build and analyze a broader class of models.

References

[1] N. Bao an J. Musacchio, "Optimizing the Decision to Expel Attackers from an Information System," Proceedings of the 47th Annual Allerton Conference on Communication and Control, Monticello, IL, Sept. 2009.

[2] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Athena Scientific, 2001, vol. I, II.

[3] N. Bao, O. P. Kreidl, and J. Musacchio, "A Network Security Classification Game," in *GameNets*, April 2011.

[4] A. Wald, *Sequential Analysis*, Wiley, New York, NY: 1947.

[5] L. Kaebling, M. Littman and A. Cassandra. "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, 101:99-134, 1998.

[6] G. Casella and R. Berger, *Statistical Inference; 2nd ed.*, Duxbury Press, 2002.

[7] L. Dritsoula, P. Loiseau, J. Musacchio, "A Game-Theoretical Approach for Finding Optimal Strategies in an Intruder Classification Game," Conference on Decision and Control, Wailea, HI, Dec. 2012.

[8] A. Gueye, J. C. Walrand, and V. Anantharam, "A Network Topology Design Game: How to Choose Communication Links in an Adversarial Environment?," in *GameNets*, April 2011.

[9] Gambit, "Gambit game theory analysis software and tools", http://www.hss.caltech.edu/gambit, 2002.

[10] L. Dritsoula, P. Loiseau, J. Musacchio, "Computing the Nash Equilibria of Intruder Classification Games," Conference on Decision and Game Theory for Security, Budapest, Hungary, Nov. 2012.